

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-167134

(43)Date of publication of application : 15.06.1992

(51)Int.Cl.

G06F 9/305

(21)Application number : 02-294717

(71)Applicant : FUJITSU LTD

(22)Date of filing : 31.10.1990

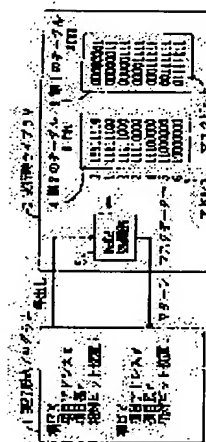
(72)Inventor : KIMURA SHIGERU

## (54) BIT TRANSCRIPTION SYSTEM

## (57)Abstract:

PURPOSE: To transcribe the bits at a high speed by attaining a bit transcription system where the mask data is generated from a data table storing the least quantity of mask data and the transcribing and transcribed areas are cleared.

CONSTITUTION: A 1st table 3 consists of plural mask data of the processing unit width which contain the mask bits that mask the areas out of a cleared area and are continuous from the lower place side. A 2nd table 4 consists of plural mask data of the processing unit width which contain the mask bits continuous from the higher place side. A transcription processing part 5 defines the data obtained by applying an AND to a pair of mask data on one of tables 3 and 4 or a pair of mask data on both tables respectively as the 1st and 2nd mask data in accordance with the relative bit position and the bit length of a transcribing area. Then the part 5 clears the data on both tables and performs an AND between both cleared data. Thus the mask data are generated from the least quantity of data that contains the mask bits continuous from both boundaries. As a result, the bit transcribing speed is extremely increased.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平4-167134

⑬ Int. Cl.<sup>3</sup>

識別記号

庁内整理番号

⑭ 公開 平成4年(1992)6月15日

G 06 F 9/305

9189-5B

G 06 F 9/30

3 4 0 A

審査請求 未請求 請求項の数 1 (全7頁)

⑮ 発明の名称 ビット転記方式

⑯ 特 願 平2-294717

⑰ 出 願 平2(1990)10月31日

⑱ 発 明 者 木 村 茂 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内

⑲ 出 願 人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

⑳ 代 理 人 弁理士 井 桁 貞一

明 細 書

1. 発明の名称

ビット転記方式

2. 特許請求の範囲

処理単位の第1の領域のうちの相対ビット位置およびビット長で示される転記元領域をマスクして他の領域をクリアする第1のマスクデータと、処理単位の第2の領域のデータのうちの他の領域をマスクして転記先領域をクリアする第2のマスクデータとを生成し、第1および第2の領域の前記領域をクリアした後論理和して第1の領域から第2の領域の対応する該転記先領域にデータを転記するビット転記方式であって、

クリア領域外をマスクするマスクビットが下位側から連続する処理単位幅の複数のマスクデータから成る第1のテーブル(3)と、

該マスクビットが上位側から連続する処理単位幅の複数のマスクデータから成る第2のテーブル(4)と、

前記相対ビット位置およびビット長に応じ、第1および第2のテーブルのうちのいずれか一方のテーブルの1組のマスクデータ、または両方のテーブルの各1組のマスクデータを論理積したデータをそれぞれ第1のマスクデータとし、他方のテーブルの1組のマスクデータ、または両方のテーブルの各1組のマスクデータを論理和したデータをそれぞれ第2のマスクデータとして前記クリア処理を行う転記処理部(5)とを設け、

転記元領域の相対ビット位置およびビット長に基づき、下位側および上位側からマスクビットが連続する処理単位幅の複数のマスクデータにより第1および第2のマスクデータを生成して転記処理を行うことを特徴とするビット転記方式。

3. 発明の詳細な説明

(概 要)

本発明はビット転記方式に関し、

転記処理を高速化することを目的とし、

処理単位の第1の領域のうちの相対ビット位置

およびビット長で示される転記元領域をマスクして他の領域をクリアする第1のマスクデータと、処理単位の第2の領域のデータのうちの他の領域をマスクして転記先領域をクリアする第2のマスクデータとを生成し、第1および第2の領域の前記領域をクリアした後論理和して第1の領域から第2の領域の対応する該転記先領域にデータを転記するビット転記方式であって、クリア領域外をマスクするマスクビットが下位側から連続する処理単位幅の複数のマスクデータから成る第1のテーブルと、該マスクビットが上位側から連続する処理単位幅の複数のマスクデータから成る第2のテーブルと、前記相対ビット位置およびビット長に応じ、第1および第2のテーブルのうちのいずれか一方のテーブルの1組のマスクデータ、または両方のテーブルの各1組のマスクデータを論理積したデータをそれぞれ第1のマスクデータとし、他方のテーブルの1組のマスクデータ、または両方のテーブルの各1組のマスクデータを論理和したデータをそれぞれ第2のマスクデータとして前

記クリア処理を行う転記処理部とを設け、転記元領域の相対ビット位置およびビット長に基づき、下位側および上位側からマスクビットが連続する処理単位幅の複数のマスクデータにより第1および第2のマスクデータを生成して転記処理を行うように構成する。

#### 〔産業上の利用分野〕

本発明は、COBOL言語等で作成されたプログラムを実行する際、呼出されて転記処理を行う実行時ライブラリにおけるビット転記方式の改良に関する。

COBOL言語等では、ハードウェアレベルの情報であるビットを操作する言語仕様が強化されつつある。

ビットを操作する項目として、ハードウェアの処理単位（例えばバイト単位）で扱う外部プール項目と、ビットそのものを扱う内部プール項目とがあるが、項目間の転記処理は内部プール項目の方がメモリが少なくすむため、内部プール項目

として扱う方が主流となっている。

しかし、内部プール項目の転記処理は外部プール項目のそれよりも処理速度が遅く、より高速な転記処理方式が求められている。

#### 〔従来の技術〕

第7図はプログラム構成図、第8図は内部プール項目の転記処理説明図、第9図は従来の転記処理フローチャート図である。

COBOLのソースプログラムで内部プール項目Xから内部プール項目Yへの転記は、

X PIC 1(2)

Y PIC 1(2)

MOVE X to Y

のように記述される。ここで(2)は項目長である。

このビット転記処理は、第7図に示すように、通常、実行時に呼出される実行時ライブラリ2で処理されており、コンパイラは、上記プログラムを、第8図に示すように、プロセッサの処理単位

の領域（例えば1バイト）のアドレス（項目アドレス）x、y、項目長n、相対ビット位置i、j（ここでは下位側からの先頭ビット位置）を設定し（第8図参照）、転記処理部10を呼び出す実行形式プログラム1に翻訳する。

領域x、yには他の項目が格納されている場合があり、呼出された転記処理部10は、指定された領域x、y内の他のデータを保存しつつ転記処理を行う。このため、

- (1) 領域xの内容をワーク領域x'に転記し、領域x'のうち、相対ビット位置i、項目長nで指定された領域を除く他の領域をクリアし、
- (2) 転記先の相対ビット位置jに合わせ、
- (3) 領域yのうち転記先領域のみをクリアし、
- (3) (2)と(3)の結果を論理和する

ことにより、領域xの転記対象データを領域yに転記する。

第9図は、以上の処理を実現する従来の転記処理例を示したものである。即ち、

- ① x領域の内容をワーク領域x'に転記する。

- ② 指定された項目のうちの1ビットに着目し、他のビットをクリアする。 $\Rightarrow x'$
- ③ 転記元と転送先の相対ビット位置  $i, j$  を比較してシフト要か否かをチェックする。
- ④ シフト要ならば、転記先の相対ビット位置  $j$  に合わせるため、領域  $x'$  内を1ビットづつシフトする。
- ⑤ マスクデータを生成する。これは、例えば、ワーク領域  $a$  に (01) (16進表示) を書き込み、 $(i-j)$  ビット分左にシフトした後、1の補数をとる等の方法により生成する。
- ⑥ マスクデータと領域  $y$  と論理積する。 $\Rightarrow y$
- ⑦ 領域  $x'$  と領域  $y$  のデータを論理和する。
- ⑦の結果を  $y$  に格納し、次のビットに着目して④～⑦の処理を繰り返す。

以上のごとく、転送元または転送先が2バイトにまたがる場合に対処できるよう、1ビットづつ転記処理を行っている。

幅の複数のマスクデータから構成される。

4は第2のテーブルで、マスクビットが上位側から連続する処理単位幅の複数のマスクデータから構成される。

5は転記処理部で、転記元領域の相対ビット位置およびビット長に応じ、第1のテーブル3および第2のテーブル4のうちのいずれか一方のテーブルの1組のマスクデータ、または両方のテーブルの各1組のマスクデータを論理積したデータをそれぞれ第1のマスクデータとし、他方のテーブルの1組のマスクデータ、または両方のテーブルの各1組のマスクデータを論理和したデータをそれぞれ第2のマスクデータとしてクリア処理を行い、クリア処理した両方のデータを論理和することにより転記する。

#### (作用)

転記元領域と転記先領域のビット位置が対応する状態において、転記処理部5は、一方の境界からマスクビット"1"が連続する複数のマスクデー

(発明が解決しようとする課題)

以上のごとく、従来では、1ビットづつ転記処理を行っているため、処理時間がかかるという課題がある。

この1ビットづつ処理するのは、コンパイラが領域  $x, y$  として、いずれか一方または双方が2バイトにまたがって指定する場合があり、これらを同一のアリゴリズムで対処できることと、複数ビットのマスクデータを生成するためには多くの処理ステップ数が必要であるという理由による。

本発明は、上記課題に鑑み、ビット転記処理を高速化するビット転記方式を提供することを目的とする。

(課題を解決するための手段)

第1図実施例の構成図および第3図本発明のマスクデータ生成説明図より対応する機能部分を抽出して説明する。

3は第1のテーブルで、クリア領域外をマスクするマスクビットが下位側から連続する処理単位

タから成る第1のテーブル(BITR)3と、他方の境界からマスクビット"1"が連続する複数のマスクデータから成る第2のテーブル(BITL)4から、いずれか一方のテーブル、または両方のテーブルからマスクデータを読み出し、組み合わせにより第1のマスクデータおよび第2のマスクデータを生成する。

転記元領域が第1の領域の上位側または下位側境界に接する場合、両境界とも接しない場合の3つのケースがあり、以下第3図を参照しつつ、説明する。

なお、処理単位幅を8ビットとして図示領域の右側を下位側、左側を上位側とし、第1のテーブルBITR3は下位側よりマスクビット"1"が連続するマスクデータ、第2のテーブルBITL4は上位側よりマスクビット"1"が連続するマスクデータをそれぞれ格納するものとし、またビット番号は下位側から昇順に0～7とする。また  $i$  は第1の領域(または第2の領域)における相対ビット位置(先頭ビット位置、左端を先頭、右端を最終とす

る)、 $n$ はビット長(項目長)である。

(ケースⅠ) 転記元領域の最終ビット位置 $k$ が第1の領域の右側境界に一致する場合( $k = i - n + 1 = 0$ )

第1のテーブルBITR3から先頭ビット位置 $i$ とマスクビットの先頭位置 $i'$ が一致するマスクデータ(第1図のテーブル配置ではアドレス $i$ のマスクデータ、BITR( $i$ )と表す)を讀出して第1のマスクデータとし、第2のテーブルBITL4から(先頭ビット位置 $i$ ) + 1とマスクビットの最終位置 $k'$ が一致するマスクデータ(BITL( $i$ ))を讀出して第2のマスクデータとする。

(ケースⅡ) 転記元領域の先頭ビット位置 $i$ が第1の領域の左側境界と一致する場合( $i = 7$ )

第2のテーブルBITL4から転送元領域の最終ビット位置 $k$ とマスクビットの最終位置 $k'$ が一致するマスクデータ(BITL( $7-n$ ))を讀出して第1のマスクデータとし、第1のテーブルBITR3から(最終ビット位置 $k$ ) - 1とマスクビットの先頭位置 $i'$ が一致するマスクデータ(BITR( $7-n$ )))

論理和する。

以上のごとく、両境界からそれぞれマスクビットが連続する最小限のマスクデータにより第1および第2のマスクデータを生成するため、1ビットずつ転送処理を行う場合に比較して大幅に転記処理を高速化することが可能となる。

#### (実施例)

本発明の実施例を図を用いて詳細に説明する。

第1図は実施例の構成図、第2図は本発明のマスクデータ生成説明図、第3図は実施例の転記処理フローチャート図、第4図は処理Aフローチャート図、第5図は処理Bフローチャート図、第6図は処理Cフローチャート図である。

以下、処理単位を8ビットとし、従来例と同一条件として説明する。

第1のマスクテーブルBITR3は、下位側境界よりマスクビット"1"がそれぞれ1, 2, ..., 7個連続し、それ以外のビットが"0"の7組のマスクデータから構成され、第1図のごとく配列されて

を讀出して第2のマスクデータとする。

(ケースⅢ) 転記元領域がどちらの境界にも接しない場合

第2のテーブルBITL4のうちの最終ビット位置 $k$ とマスクビットの最終位置 $k'$ が一致するマスクデータ(BITL( $i-n$ )))と、第1のテーブルBITR3のうちの先頭ビット位置 $i$ とマスクビットの先頭位置 $i'$ が一致するマスクデータ(BITR( $i$ )))とを論理積したデータを第1のマスクデータとし、第1のテーブルBITR3のうちの(最終ビット位置 $k$ ) + 1とマスクビットの先頭位置 $i'$ が一致するマスクデータ(BITR( $i-n$ )))と、第2のテーブルBITL4のうちの(先頭ビット位置 $i$ ) + 1とマスクビットの最終位置 $k'$ が一致するマスクデータ(BITL( $i$ )))とを論理和したデータを第2のマスクデータとする。

そして、転記処理部3は、第1のマスクデータと第1の領域のデータとを論理積し、第2のマスクデータと第2の領域のデータとを論理積してクリア処理を行った後、クリア処理した両データを

いる。

第2のマスクテーブルBITL4は、上位側境界よりマスクビット"1"がそれぞれ1~7個連続し、それ以外は"0"の7組のマスクデータから構成され、第1図のごとく配列されている。

転記処理部5は、転記元および転記先の設定条件である項目アドレス $x$ ,  $y$ , 項目長 $n$ , 相対ビット位置 $i$ ,  $j$ が与えられて呼出されたとき、第1および第2のマスクテーブルBITR3, BITL4から転記処理に必要な第1および第2のマスクデータを以下に示す処理によって生成し、転記処理を行う。

第2図は、転記データの先頭ビットが上位側境界と一致する場合( $i = 7$ )、転記データの最終ビットが下位側境界と一致する場合( $i - n + 1 = 0$ )およびいずれも一致しない場合を相対ビット位置 $i$ および項目長 $n$ により判別してそれぞれの処理を行うように構成したものである。即ち、(1) 転記元領域と転記先領域のビット位置が異なる( $i \neq j$ )か否かを判別し、異なる場合は処理

Dに進む。

(2)  $i = j$ で、且つ  $i = 7$  ならば転記元領域が上位側境界に一致するから処理Aに進む。

(3)  $i = j$ で、且つ  $j \neq 7$  ならば、最終ビット位置をチェックする。即ち、 $i - n + 1 = 0$  ならば最終ビットが下位側境界と一致するから、処理Bに進む。

(4)  $i - n + 1 \neq 0$  ならば、処理Cに進む。

以下処理A～処理Dの詳細を説明するが、領域  $x$  の他のビットを保存するため、すべての処理で領域  $x$  のデータをワーク領域  $x'$  (第1の領域に対応する) に転記して(処理A～Cのステップ①)領域  $x'$  から領域  $y$  (第2の領域に対応する) への転記処理を行う。

(処理A) 第4図参照

前述の(ケースⅡ)に相当するもので、 $(7 - n)$  をアドレスとして第2のテーブルBITL4 および第1のテーブルBITR3 からマスクデータを読み出し、それぞれ第1および第2のマスクデータとする。第4図は処理ステップを示したもので、

$$\textcircled{2} (x') \cdot \text{BITL}(7-n) \Rightarrow x'$$

この論理積演算により、領域  $x$  のデータのうち転記元領域以外の領域がクリアされたデータが領域  $x'$  に得られる。

$$\textcircled{3} (y) \cdot \text{BITR}(7-n) \Rightarrow y$$

これにより領域  $y$  のうち転記先領域のみクリアされたデータが領域  $y$  に得られる。

$$\textcircled{4} (x') + (y) \Rightarrow y$$

これにより、転記元領域のデータが領域  $y$  に転記される。

(処理B) 第5図参照

前述の(ケースⅠ)に相当する。(i) をアドレスとして第1のテーブルBITR3 および第2のテーブルBITL4 からマスクデータを読み出し、処理Aと同様に以下の演算処理を行う。

$$\textcircled{2} (x') \cdot \text{BITR}(i) \Rightarrow x'$$

$$\textcircled{3} (y) \cdot \text{BITR}(i) \Rightarrow y$$

$$\textcircled{4} (x') + (y) \Rightarrow y$$

これにより、転記元領域のデータが領域  $y$  に転記される。

(処理C) 第6図参照

(ケースⅢ)に相当するもので、第6図は第1および第2のマスクデータを生成する代わりに、第1および第2のテーブルから読み出したそれぞれのマスクデータにより直接  $(x')$ 、 $(y)$  に処理を施した例を示したものである。即ち、

$(\text{BITR}(i) \cdot (x')) \cdot \{\text{BITL}(i-n) \cdot (x')\}$  を演算し、

$(\text{BITL}(i) \cdot (y')) + \{\text{BITR}(i-n) \cdot (y')\}$  を演算し、両者の結果を論理和している。

(処理D)

$i \neq j$  の場合、図示省略したが、領域  $x'$  をシフトして転記先に合わせ、処理A～処理Cを適用する。なお、この場合は従来の処理を行ってもよい。

以上の処理で、項目が1処理単位の領域に限ったが、2領域にまたがる場合は、それぞれの領域で処理A～処理Cを適用すればよい。

以上のごとく、2組のテーブルよりマスクデータを生成してクリア処理を施し転記処理を行うた

め、項目長が複数ビットの場合でも一度の処理で行うことができ、また、マスクデータの生成が簡単になるため、転記処理の高速化が達成される。

(発明の効果)

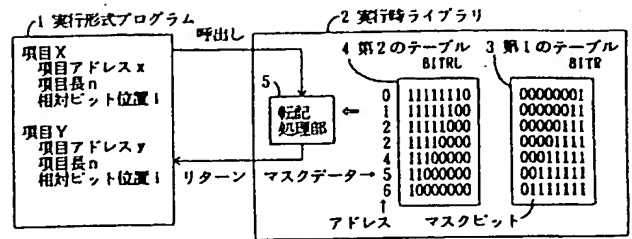
以上説明したように、本発明は、最小限のマスクデータを格納したテーブルからマスクデータを生成して、転記元および転記先領域のクリア処理を行うビット転記方式を提供するもので、ビット転記処理の高速化に多大な効果を奏する。

#### 4. 図面の簡単な説明

第1図は実施例の構成図、第2図は実施例の転記処理フローチャート図、第3図は本発明のマスクデータ生成説明図、第4図は処理Aフローチャート図、第5図は処理Bフローチャート図、第6図は処理Cフローチャート図、第7図はプログラム構成図、第8図は内部ブール項目の転記処理説明図、第9図は従来の転記処理フローチャート図である。

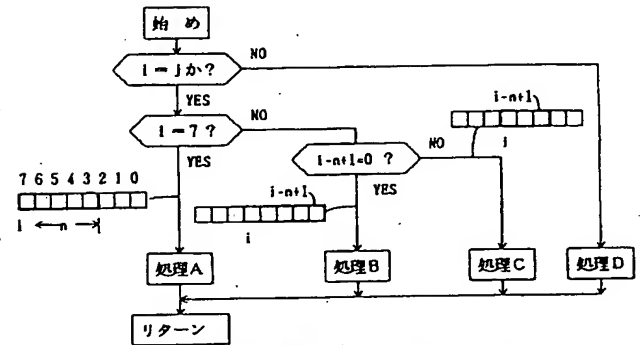
図中、1 は実行形式プログラム、2 は実行時ライブラリ、3 は第1のテーブルBITR、4 は第2のテーブルBITL、5、10 は転記処理部である。

代理人 弁理士 井 術 貞一



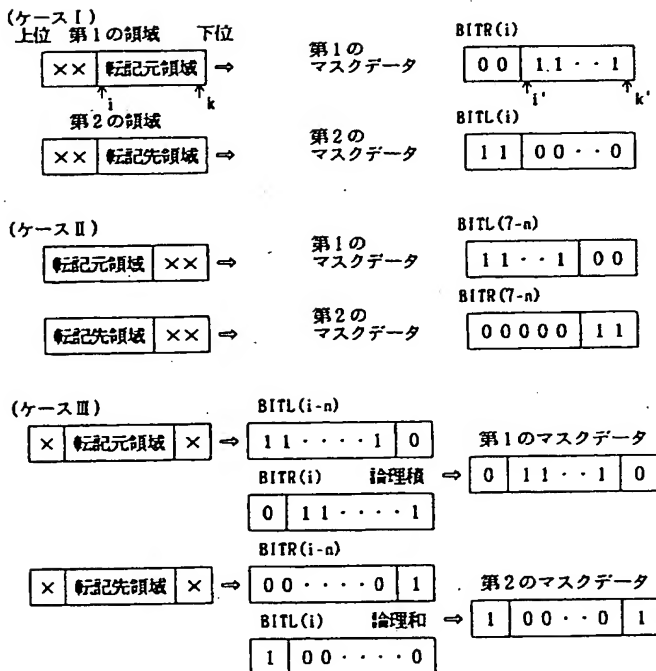
実施例の構成図

第 1 図



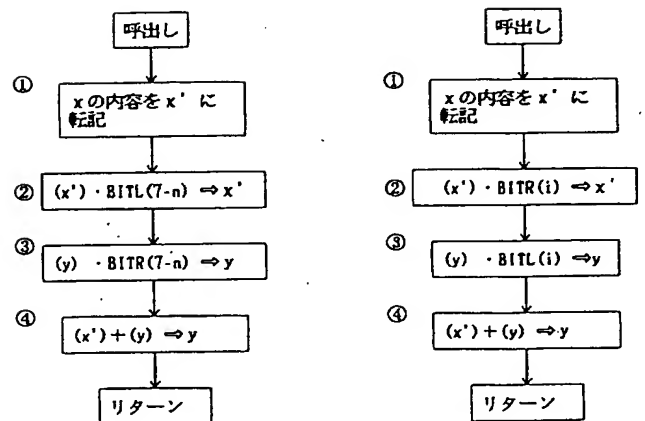
実施例の転記処理フローチャート図

第 2 図



本発明のマスクデータ生成説明図

第 3 図



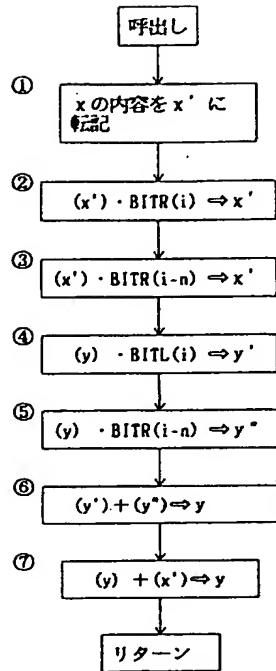
処理Aフローチャート図

第 4 図

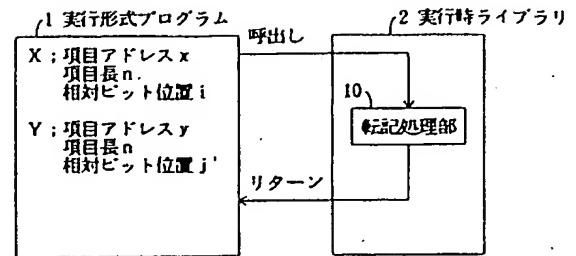
処理Bフローチャート図

第 5 図

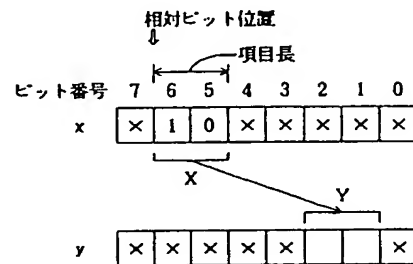




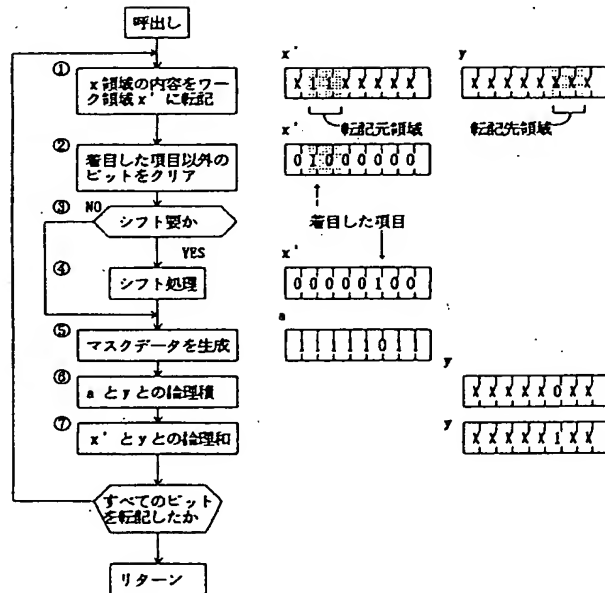
処理Cフローチャート図  
第 6 図



プログラム構成図  
第 7 図



内部プール項目の転記処理説明図  
第 8 図



従来の転記処理フローチャート図  
第 9 図